TARAD: Task-Aware Robot Affordance-Centric Diffusion Policy Learned From LLM-Generated Demonstrations

Site Hu, Takayuki Nagai, and Takato Horii

***Abstract*—In open-ended task settings, the ability of a robot to execute diverse tasks accurately by following language instructions is critical. Methods based on traditional imitation learning typically depend on extensive expert demonstrations and often struggle to generalize in the case of unseen scenarios or tasks. Recently, approaches leveraging large foundational models have demonstrated improved generalization by enhancing task comprehension in novel scenarios based on the intrinsic world knowledge embedded in these models. However, these methods rely on predefined motion primitives and lack a detailed understanding of the environment, which is essential for successful execution. Herein we introduce Task-Aware Robot Affordance-Centric Diffusion Policy (TARAD), a novel framework for robot manipulation. TARAD leverages large language models and vision-language models to perform high-level planning from natural language instructions and extract affordance information from the robot's observations. A heuristic motion planner is employed for low-level motion planning, enabling zero-shot trajectory synthesis and the fully automatic generation of a dataset with language labels and affordances. By incorporating affordances into the observation space, our approach integrates the intrinsic commonsense and reasoning capabilities of foundation models into imitation learning, enabling the training of an affordance-centric, multi-task three-dimensional (3D) diffusion policy. Empirical evaluations in both the RLBench simulated environments and real-world experiments with UR5e demonstrate that TARAD effectively combines the precise control of imitation learning with the strong generalization capabilities of foundation models, all without relying on expert demonstrations or predefined motion primitives.**

***Index Terms*—AI-enabled robotics, learning from demonstration, manipulation planning.**

I. Introduction

GENERAL-PURPOSE robot manipulation learning faces the critical challenge of understanding natural language instructions to execute a wide range of tasks accurately in real-world settings while effectively generalizing across diverse environments [1]. Conventional approaches based on imitation learning typically require numerous expert demonstrations [2], [3] and tend to overfit to specific tasks [4], thereby limiting their practical applicability to unseen scenarios.

Recently, researchers have explored the use of affordances to enhance robotic interaction in unstructured environments [5], [6]. However, extending affordance learning from manually annotated datasets to open-world settings with arbitrary natural language instructions remains a challenge [7]. With the rapid advancement of large language models (LLMs) exhibiting strong generalization capabilities [8], researchers are increasingly integrating foundation models trained on Internet-scale data into robotic systems. Typically, these methods involve decomposing natural language instructions into high-level plans using LLMs and interpreting the environment via perception APIs or textual scene descriptions, while relying on predefined motion primitives for low-level control [9], [10], [11]. Unfortunately, such approaches often fail to capture the fine-grained affordances necessary for precise task execution, leading to task failures even with correct high-level planning [12]. In recent studies, learning-based policies have been combined with foundation models to achieve both precise and generalized manipulation; however, these methods still depend on expert demonstrations [1] or predefined motion primitives [13].

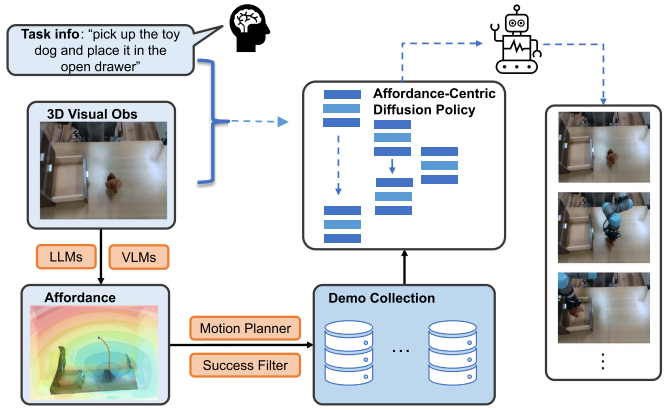

Fig. 1. TARAD autonomously collects demonstrations by extracting language-conditioned affordances from RGB-D images to train an affordance-centric diffusion policy.

Received 19 March 2025; accepted 2 August 2025. Date of publication 14 August 2025; date of current version 25 August 2025. This article was recommended for publication by Associate Editor W. Yu and Editor A. Faust upon evaluation of the reviewers' comments. This work was supported by the New Energy and Industrial Technology Development Organization (NEDO) and JST Moonshot R&D Grant. Number under Grant JPMJMS2011. *(Corresponding author: Site Hu.)*

Site Hu and Takato Horii are with the Department of Systems Innovation, Graduate School of Engineering Science, Osaka University, Osaka 565-0871, Japan (e-mail: s.hu@rlg.sys.es.osaka-u.ac.jp; takato@sys.es.osaka-u.ac.jp).

Takayuki Nagai, deceased, was with the Department of Systems Innovation, Graduate School of Engineering Science, Osaka University, Osaka 565-0871, Japan.

Digital Object Identifier 10.1109/LRA.2025.3598998

To overcome these limitations, we propose **TARAD** (**T**ask-**A**ware **R**obot **A**ffordance-centric **D**iffusion Policy), a novel framework for robot manipulation. As shown in Fig. 1, TARAD leverages LLMs for high-level planning from natural language instructions and employs both LLMs and vision-language models (VLMs) to extract affordance information from robot observations. The extracted affordances are represented as point clouds and voxel-based value maps [14] and are input into a heuristic motion planner, which automatically generates a dataset annotated with both affordance and language labels. By integrating affordances into the observation space, our approach embeds the intrinsic commonsense and reasoning capabilities of foundation models within an imitation learning-based framework, thereby training an affordance-centric, multi-task three-dimensional (3D) diffusion policy.

Our contributions can be summarized as follows:

- We introduce a novel language-guided data collection framework for robust synthesis of trajectories without relying on predefined motion primitives and automatic generation of a dataset annotated with language and affordance information.
- We integrate affordance information into a 3D diffusion policy to achieve robust generalization across tasks and environments.
- We evaluate TARAD on eight simulated tasks in RLBench [15] and three real-world tasks, demonstrating that TARAD effectively combines the precise control of imitation learning with the strong generalization capabilities of foundation models without the need for expert demonstrations or predefined motion primitives.

II. Related Works

A. Robot Learning for Manipulation

Recent approaches to robot manipulation learning have predominantly relied on reinforcement learning [16] and imitation learning [17] to derive effective policies from expert demonstrations using images or point clouds as observation spaces [17], [18]. Furthermore, diffusion models [19] have recently been employed to handle multi-modal actions [20], [21], [22], significantly enhancing policy adaptability and robustness. However, these methods are especially effective for tasks similar to those encountered during training and struggle to generalized to novel tasks [23]. In addition, these methods often lack natural language understanding and generalization abilities. Chen et al. [1] addressed this challenge by leveraging VLMs to generate spatial value maps that guide action diffusion. However, the acquisition of sufficient and diverse data for robot learning remains a challenge. By contrast, our method leverages LLMs as demonstration generators and integrates both LLMs and VLMs for affordance extraction, effectively embedding the strong generalization capabilities of foundation models into a diffusion-based policy framework.

B. Affordance for Robotics

Affordance [24] refers to the action possibilities that an actor can readily perceive [7]. In robotic systems, effective affordance extraction is crucial for enabling nuanced interactions with unstructured environments [5], [6]. Affordances are typically represented using images or point clouds. Action possibilities are often represented as affordance maps, which quantify the likelihood of executing specific actions at given locations [25]. Deep learning methods have been widely used to predict such affordances [26], [27], and recent studies have begun to extract affordance knowledge from large pre-trained models [7], [14].

C. Foundation Models for Robot Manipulation

Recent advances in LLMs [8] and VLMs [28], [29] have enabled their use in robot manipulation, typically as high-level planners for interpreting language and scene context. Low-level control often depends on motion primitives [9], [10], [11], limiting fine-grained affordance reasoning [12].

To improve spatial reasoning, recent work integrates visual feedback [30] or spatial value maps [14]. CoPa [12] combines spatial constraints from LLMs and VLM-based scene understanding to locate grasp targets, but relies on predefined grasp models and manually defined geometric assumptions. ReKep [31] generates code from keypoint constraints but requires accurate tracking and handcrafted solvers. Unlike these, we use foundamation models to collect suboptimal data [20], which is then distilled into an affordance-conditioned diffusion policy.

Foundation models have also been used to synthesize demonstrations. Gensim [32] and Gensim2 [33] generate simulated tasks and execution code, but remain limited to simulation and rely on complex prompts and predefined constraints. Ha et al. [20] use LLMs to compose predefined 6-DoF exploration primitives with a verify-and-retry loop, requiring days of simulation to collect data for training image-conditioned diffusion policies. Similarly, Jin et al. [13] use LLMs to sequence predefined robot skills to obtain data for training. In contrast, we extract voxel-level affordances using LLMs and VLMs, enabling efficient demonstration synthesis via a simple heuristic planner. Our affordance-conditioned diffusion policy shares representations across data generation and learning, directly distilling structured knowledge from foundation models. The same pipeline runs on real robots with minimal modification, eliminating the sim-to-real gap faced by prior works [13], [20].

III. Method

In this work, we propose a novel framework that generates robot manipulation trajectories solely from natural language task descriptions without relying on predefined motion primitives. Our method automatically constructs a dataset annotated with language labels and affordance information, and leverages this dataset to train a affordance-centric diffusion policy with strong generalization capabilities. As illustrated in Fig. 2, the framework comprises two main phases: data generation and policy training. During the data generation phase, natural language instructions are decomposed into high-level task plans using LLMs (Sec. III-A). Subsequently, we use both LLMs and VLMs to extract the affordances (Sec. III-B), represented by target-object point clouds and voxel value maps introduced

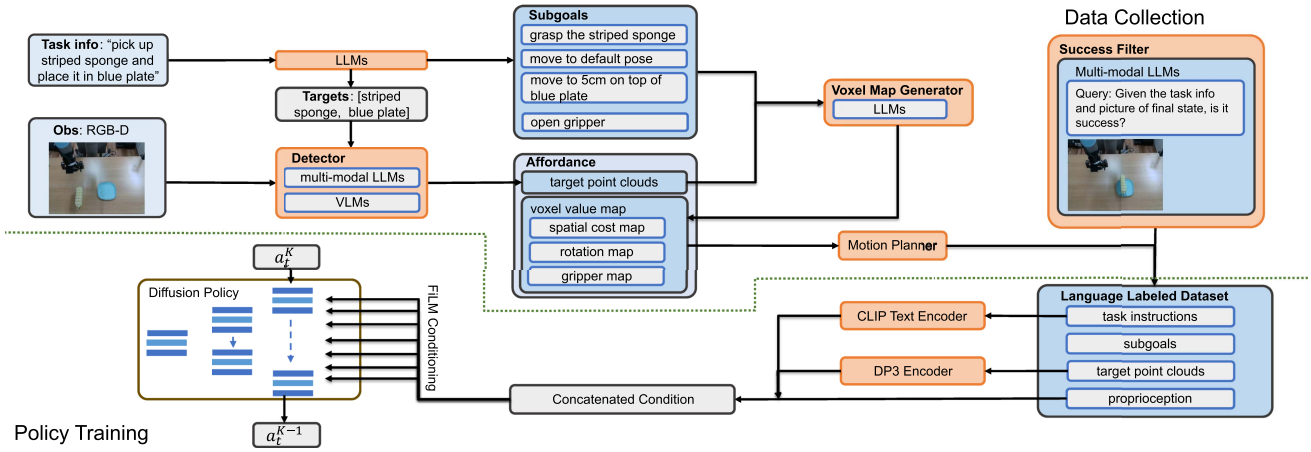

Fig. 2. Overview of our framework, which proceeds in two main phases: data generation and policy training. In the data generation phase, LLMs decompose instructions into high-level task plans, while LLMs and VLMs jointly extract affordance based on high-level plans, represented as target object point clouds and voxel value maps. A heuristic motion planner then translates these high-level plans into low-level executions, and successful trials are automatically filtered and labeled for dataset collection. In the subsequent policy training phase, the affordance representations form the observation space for a multi-task vision-language conditional diffusion model, enabling robust task generalization.

in [14]. Based on these affordances, high-level plans are translated into low-level motions using a heuristic motion planner. Meanwhile, both LLMs and VLMs are leveraged to filter out successful executions, which will be collected as labeled datasets (Sec. III-C). In the policy training phase (Sec. III-D), affordance representations serve as the observation space for a multi-task vision-language conditional diffusion model, enabling robust generalization across multiple tasks.

A. Task Decomposition

Given a natural language task description, we use the advanced large language model GPT-4o [8] to generate a structured high-level task plan. The model decomposes instructions into a sequence of sub-goals to capture semantic affordance and task dependencies, thereby facilitating effective execution.

B. Affordance Extraction

Building on the high-level plan, our approach extracts affordance information using both LLMs and VLMs. We follow the prompting structure from [14], recursively call the perception module using their own generated code. Affordance is represented as target object point clouds and the voxel value maps, which consist of three components: (i) a spatial cost map m_c that assigns lower costs near the target object and higher costs further away, (ii) an end-effector orientation map m_r that specifies the required end-effector orientation, and (iii) a gripper map m_g that indicates the appropriate gripper actions.

Specifically, we employ GPT-4o to extract target object names from the task description, utilize the open-vocabulary detector GroundingDINO [28] to predict bounding boxes from RGB-D observations, and then leverage the multi-modal capabilities of GPT-4o to verify and refine the predicted bounding boxes and enhance the recognition accuracy. Subsequently, we apply the Segment Anything Model 2 [29] to obtain and track segmentation masks for extracting the target object point clouds. An LLM-generated script then computes the voxel value maps. The entire voxel value maps calculation process is detailed in Alg. 1, adapted from [14].

C. Low-Level Motion Execution

Once the voxel value maps are obtained, a heuristic motion planner converts the high-level plan into low-level motions. A greedy search in the voxel space identifies a trajectory that minimizes the cumulative spatial cost. At each point along the trajectory, the corresponding end-effector orientations and the gripper actions derived from the orientation and gripper maps are integrated to construct a low-level motion plan. A visualization of this process in a real-world environment is shown in Fig. 3. After executing the plan, we use GPT-4o to evaluate whether the task is successful. All successful trajectories, along with the corresponding low-level motion commands, robot proprioception data, target object point clouds, RGB-D observations, linguistic task descriptions, and sub-goals, are automatically stored to create an annotated dataset. This automated pipeline ensures that the collected demonstrations are both physically feasible and semantically aligned with the task descriptions.

D. Affordance-Centric Diffusion Policy Distillation

Using the automatically collected dataset enriched with language labels and affordance information, we train a conditional diffusion model [34] to learn a multi-task robot manipulation policy. The diffusion model is designed to predict the noise $\epsilon_\theta(a_k, k, c)$ added at each diffusion time step k, and then iteratively denoise a random Gaussian noise vector into the desired action via a reverse diffusion process. Specifically, we encode the affordance represented by the downsampled point clouds of the target objects into a compact 3D feature representation using a lightweight MLP encoder (DP3) [21], and encode the task instructions using the CLIP B/32 text encoder [35]. These representations, together with the proprioception history, are

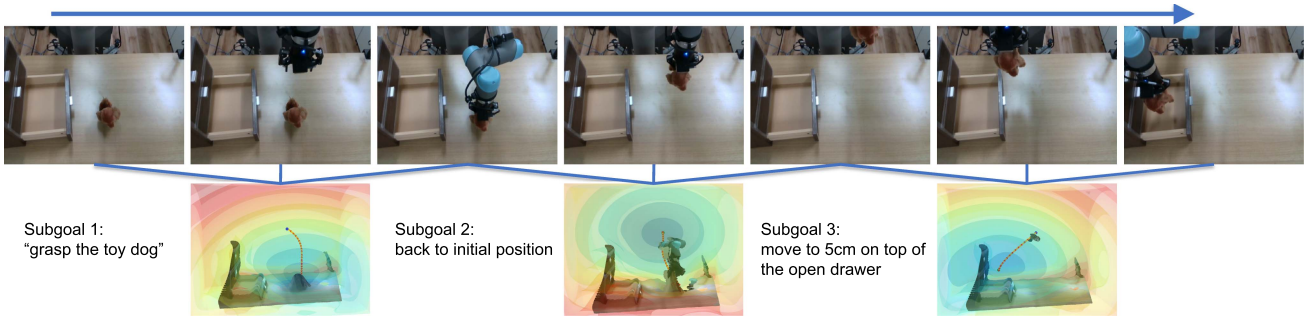

Fig. 3. Visualization of spatial cost maps and planned trajectories for **ToyInDrawer** task in real-world environment. The spatial cost maps guide grippers towards target positions derived from the LLMs and VLMs with a heuristic motion planner. Meanwhile, the orientation and gripper maps determine the gripper's rotation and state along the planned trajectories.

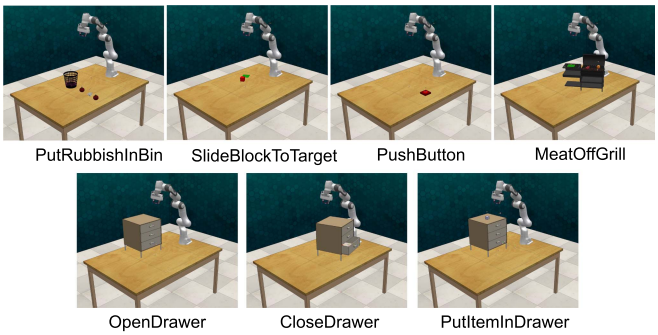

Fig. 4. Seven tasks in simulation experiment. **MultiTaskDrawer** used in our experiment is composed of **OpenDrawer**, **CloseDrawer**, and **PutItemInDrawer** tasks.

concatenated to form a composite conditioning vector c, which is incorporated via FiLM [36] during the denoising process. Here, we use the task instruction rather than subgoals as condition to avoid extra latency for monitoring subgoals.

Starting from a random Gaussian noise vector $a_K \sim \mathcal{N}(0, I)$, the denoising network e_θ is iteratively applied over K steps to yield the final action a_0. At each time step k, the reverse diffusion update is computed as follows:

$$a_{k-1} = \frac{1}{\sqrt{\alpha_k}} \left(a_k - \frac{1-\alpha_k}{\sqrt{1-\bar{\alpha}_k}} \epsilon_\theta(a_k, k, c) \right) + \sigma_k z \qquad (1)$$

where α_k is the noise schedule parameter at time step k, $\bar{\alpha}_k = \prod_{i=1}^{k} \alpha_i$, σ_k denotes the noise scale, $z \sim \mathcal{N}(0, I)$. This iterative denoising process progressively refines the noisy action vector until $k = 0$, resulting in the final action a_0.

The training objective is to minimize the mean squared error between the true noise ϵ added to the original data and the noise predicted by the model. Formally, the loss function is defined as:

$$\mathcal{L} = \mathbb{E}_{a_0, \epsilon, k} \left[\|\epsilon - \epsilon_\theta(a_k, k, c)\|^2 \right] \qquad (2)$$

During inference, the target object point cloud is extracted as described in Sec. III-B and used as input to the diffusion policy.

Algorithm 1: Voxel Value Maps Calculation.

Input: Object point clouds P, subgoal sg_i, map size S_m, $LLMs$

Output: Spatial cost map m_c, end-effector orientation map m_r, gripper map m_g

Initialize $m_c, m_g \in \mathbb{R}^{S_m \times S_m \times S_m}$

Initialize $m_r \in \mathbb{R}^{S_m \times S_m \times S_m \times 4}$

// Get target pose, gripper action and voxel radius

$(x, y, z), (r_w, r_x, r_y, r_z), \beta_r, a_g, \beta_g \leftarrow LLMs(P, sg_i)$

Get voxel coordinates $(i, j, k) \leftarrow (x, y, z)$

$m_c(u, v, w) \leftarrow \sqrt{(u-i)^2 + (v-j)^2 + (w-k)^2}$; $m_c(i, j, k) \leftarrow 0$ △ Spatial cost map

$m_r(u, v, w) \leftarrow R_{\text{init}}$; $m_g(u, v, w) \leftarrow G_{\text{init}}$ △ Init maps

for (u, v, w) in $\{(u, v, w) \mid |u-i|, |v-j|, |w-k| \leq \beta_r\}$ **do**

$m_r(u, v, w) \leftarrow (r_w, r_x, r_y, r_z)$

end for

for (u, v, w) in $\{(u, v, w) \mid |u-i|, |v-j|, |w-k| \leq \beta_g\}$ **do**

$m_g(u, v, w) \leftarrow a_g$

end for

return (m_c, m_r, m_g)

IV. Experiments

In this section, the proposed system is evaluated in both simulation and real-world environments. Our experiments aim to address the following questions:

1) Can TARAD's data collection method effectively gather a dataset annotated with language labels and affordance information solely from natural language instructions?
2) Can TARAD's policy learning approach distill an effective visuo-linguo-action policy from the collected dataset?
3) Does TARAD exhibit strong generalization to new object instances, unseen scenes, and different views?

A. Experiment Setup

In the simulation, we utilize the RLBench environments [15], which provide five cameras and oracle segmentation masks for each object. We evaluate our system on three single-task

TABLE I
SIMULATION RESULTS

Model	PutRubbish InBin	SlideBlock ToTarget	Push Button	MeatOff Grill	Open Drawer	Close Drawer	PutItem InDrawer	MultiTask Drawer
Voxposer [14]	75.0	55.0	**100.0**	45.0	20.0	80.0	30.0	45.0
Act3D [37]	82.8 ± 2.55	71.1 ± 2.55	89.4 ± 0.96	61.1 ± 2.55	68.9 ± 1.92	65.6 ± 3.47	57.8 ± 2.55	50.6 ± 4.20
3D Diffuser Actor [22]	**90.6** ± 3.47	81.7 ± 1.67	95.0 ± 2.89	**78.3** ± 4.41	**84.4** ± 2.55	88.3 ± 1.67	**82.2** ± 0.96	**78.9** ± 4.19
3D Diffusion Policy [21]	89.4 ± 3.85	82.8 ± 0.96	95.0 ± 3.33	61.7 ± 6.01	72.8 ± 8.22	82.2 ± 4.81	61.1 ± 7.51	66.1 ± 8.55
Ours	88.9 ± 3.47	**85.0** ± 2.89	97.8 ± 1.92	73.3 ± 3.33	78.3 ± 4.41	**92.8** ± 5.36	73.9 ± 6.31	67.2 ± 3.47

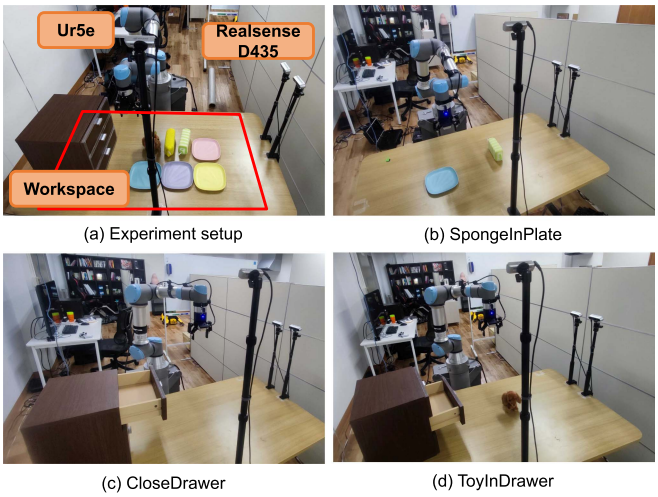

Fig. 5. Real-world experiment setup. Only one camera is used for each task. **SpongeInPlate**: Grasp a sponge and place it in a plate. **CloseDrawer**: Close one of three drawers. **ToyInDrawer**: Grasp a toy dog and place it into an open drawer.

domains, four multi-task domains, and a composite multi-task domain. In the simulation experiments, oracle masks are used instead of VLMs to extract affordance information. Fig. 4 illustrates the experimental setup in the simulation environments.

We also validate TARAD on real-world manipulation tasks using a UR5e robotic arm equipped with a RealSense D435 camera for RGB-D image capture. The evaluation spans three tasks, as illustrated in Fig. 5.

B. Simulation Experiment

In the simulated environment, we bypass the use of VLMs by leveraging oracle masks to obtain affordance information. We collected 30 demonstrations for each single-task and multi-task domain, evenly distributed across all variants. For the composite task **MultiTaskDrawer**, we uses all demos from **OpenDrawer**, **CloseDrawer**, and **PutItemInDrawer**. Using these automatically collected demonstrations, we trained our language-conditioned, affordance-centric diffusion policy. We compare our method with several baselines: a foundation-model-based approach, **Voxposer** [14], and three imitation-learning-based SOTA methods for RLBench: **Act3D** [37], the enhanced language-conditioned variant of **3D Diffuser Actor** [22] and **3D Diffusion Policy** [21]. These imitation-learning-based approaches rely on 30 expert demonstrations per task, rather than using the automatically collected demonstration data employed by TARAD.

For the foundation-model-based method, 20 episodes are evaluated for each task. For the learning-based methods, we train 2000 epochs for each task with three random seeds, evaluated 20 episodes every 200 epochs, and then computed the average for the top three success rates. Table I presents quantitative results, which indicate that our method effectively collects datasets annotated with language labels and affordance information and distills a visuo-linguo-action policy with performance comparable to SOTA methods relying on expert demonstrations.

Ablation Study: We evaluate the following ablative versions of our framework: (i) SOTA imitation baselines (3D Diffuser Actor [22] and 3D Diffusion Policy [21]) trained on the dataset generated by TARAD; (ii) the affordance-centric diffusion policy retrained on expert demonstrations automatically extracted from RLBench [15]; and (iii) a language-free TARAD that conditions only on proprioception and the affordance point clouds.

Results are summarized in Table II. While the baselines trained on expert demonstrations outperform those trained on TARAD data, the latter still achieve comparable success rates, indicating that the auto-generated trajectories are of sufficient quality for policy learning. Without relying on expert demonstrations, TARAD achieves performance comparable to 3D Diffuser Actor and superior to 3D Diffusion Policy. Under identical auto-generated datasets, TARAD significantly outperforms SOTA baselines, confirming the effectiveness of its affordance-centric diffusion policy. Removing language conditioning degrades TARAD's performance on multi-variant tasks, demonstrating the utility of language instructions for multi-task generalization. Despite that, this ablative version still outperforms language-conditioned baselines trained on the same dataset, further highlighting the synergy between affordances and diffusion policies. Interestingly, TARAD trained on expert demonstrations exhibits a slight performance drop, which we attribute to a mismatch between expert motions and the affordance-centric representation expected by the model. This suggests that the current planner-and-filter pipeline produces data that are well aligned with the proposed policy architecture.

C. Real-World Experiment

To validate TARAD in real-world settings, we deployed our system on a UR5e robotic arm. In this setup, GPT-4o is used to extract target object names from task descriptions and verify the bounding boxes predicted by the open-vocabulary detector GroundingDINO [28]. Segment Anything 2 [29] is then employed to obtain and track the segmentation masks to extract the

TABLE II
ABLATION STUDY

3D Diffuser Actor(TARAD Data)	81.7 ± 4.41	72.2 ± 5.85	95.6 ± 3.47	65.6 ± 4.19	62.2 ± 0.96	80.56 ± 3.47	41.67 ± 2.89	62.2 ± 4.81
3D Diffusion Policy(TARAD Data)	77.8 ± 3.47	66.1 ± 5.85	94.4 ± 5.36	51.1 ± 4.19	55.0 ± 5.00	74.4 ± 6.94	46.1 ± 2.55	58.3 ± 4.41
Ours	**88.9** ± 3.47	**85.0** ± 2.89	**97.8** ± 1.92	**73.3** ± 3.33	**78.3** ± 4.41	**92.8** ± 5.36	**73.9** ± 6.31	**67.2** ± 3.47
Ours(Expert Demo)	85.0 ± 1.67	72.8 ± 3.47	93.9 ± 3.47	67.8 ± 2.55	72.8 ± 0.96	80.0 ± 1.67	63.9 ± 6.74	61.1 ± 5.36
Ours(w/o Language Condition)	-	-	-	62.8 ± 8.55	68.3 ± 2.89	87.2 ± 7.52	61.7 ± 5.77	55.6 ± 5.09

TABLE III
REAL-WORLD RESULTS

Model	Sponge In Plate	Close Drawer	Toy In Drawer
Voxposer	76.7	70.0	46.7
3D Diffusion Policy(TARAD Data)	83.3	66.7	73.3
Ours	**96.7**	**93.3**	**86.7**

TABLE IV
GENERALIZATION RESULTS

Model	Task	Base	App. Change	Inst. Change	Clutter Scene	View Change-1	View Change-2
3DP	SpongeInPlate	83.3	83.3	66.7	33.3	80.0	26.7
	ItemInDrawer	73.3	70.0	63.3	26.7	66.7	6.7
Our	SpongeInPlate	96.7	93.3	83.3	96.7	96.7	90.0
	ItemInDrawer	86.7	83.3	90.0	83.3	90.0	70.0

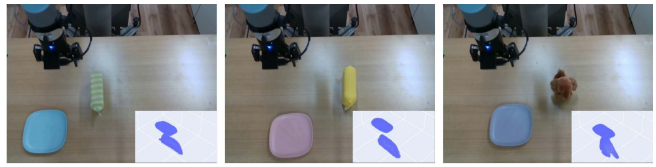

Fig. 6. Appearance and instance generalization experiments. Despite variations such as changes in sponge and plate color or replacing the sponge with a toy dog, our method effectively captures similar affordance point clouds based on the given language instruction, ensuring robust execution of the affordance-centric policy.

target object point clouds, which are used to compute the spatial cost map. By coupling VLMs with mask tracker, our method enables high-frequency perception-action loops at about 1 Hz, with 130 ms for RGB-D capture and point cloud computation, 110 ms for mask updates, and 650 ms for motion execution with the UR5e. For the SpongeInPlate task, 30 demos are collected, whereas for the CloseDrawer and ToyInDrawer tasks, 10 demos are collected per drawer. We use Voxposer [14] and 3D Diffusion Policy [21] as baselines. 3D Diffuser Actor [22] failed in our real-world settings, likely due to its high sensitivity to image inputs. Therefore, it has been omitted from the comparison for clarity and fairness. We reproduce Voxposer in our tasks by employing the same prompt templates from [14] and the same detectors used in our method. For Voxposer, we evaluate 30 episodes per task. Our method is trained for 2000 epochs, and evaluated over 30 episodes using the final checkpoint. For fair comparison, the 3D Diffusion Policy is trained on the same dataset using the same setup.

As shown in Table III, our method remains effective in real-world settings and significantly outperforms both baselines. Voxposer struggles to capture object attributes and task-specific interaction conditions, leading to poor performance on fine-grained manipulation tasks in real-world environments. Similarly, 3D Diffusion Policy lacks sufficient affordance details required for precise manipulation. By contrast, the policy distilled by TARAD enables effective and reliable object interactions.

D. Generalization Evaluation

We evaluate the generalization capability of our method and compare it with 3D Diffusion Policy [21] on two real-world tasks: SpongeInPlate and ItemInDrawer, considering four types of variations: appearance, instance, view, and scene changes. The overall results are presented in Table IV.

In the SpongeInPlate task, appearance variation is introduced by modifying the colors of the sponge and plate, whereas instance variation replaces the sponge with a toy dog. In the ItemInDrawer task, appearance variation involves changing the toy dog's color, whereas instance variation replaces the toy dog with a sponge.

For view generalization, each policy is trained using data collected from the red-circled camera and evaluated using the cameras marked by the green and blue circles, referred to as ViewChange-1 and ViewChange-2, respectively.

For scene generalization, both policies are trained in an environment free of additional objects but tested in cluttered scenes containing multiple visually similar distractors.

1) Appearance and Instance Generalization: During the data collection phase, TARAD extracts affordance information in the form of color-free target object point clouds and voxel value maps. This representation enables effective generalization across various appearances. For instance, in the SpongeInPlate task, although TARAD can automatically collect demos featuring sponges and plates of multiple colors, we train the diffusion policy using only demos with a striped sponge and a blue plate. Despite the limited training set, the policy generalizes to sponges and plates of other colors with nearly unchanged success rates. Furthermore, the use of downsampled affordance point clouds minimizes the differences between objects of similar size but varying shapes, thereby facilitating generalization to unseen object instances. As shown in Fig. 6, although our policy is trained with a sponge and a plate, it effectively captures similar affordance point clouds based on the given language instruction, ensuring robust execution of the affordance-centric policy.

2) Scene Generalization: TARAD leverages VLMs to detect target objects from RGB-D observations and employs multimodal LLMs to verify the detection results. This process enables the accurate extraction of affordance information even in

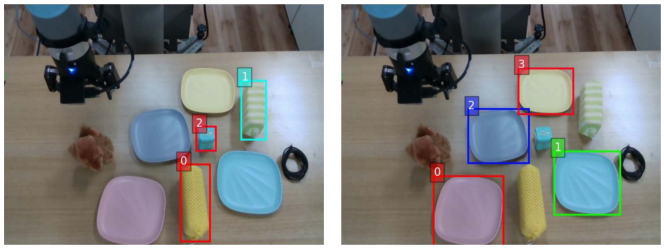

Fig. 7. Scene generalization experiments. The policy was trained in a simplified environment containing only a striped sponge and a blue plate and tested in a cluttered scene with modified language instructions: "pick up the yellow sponge and place it into the pink plate". The VLMs identified multiple similar objects in the cluttered environment, while the multimodal LLMs accurately recognized the intended target objects (both labeled as box 0) based on the updated instructions.

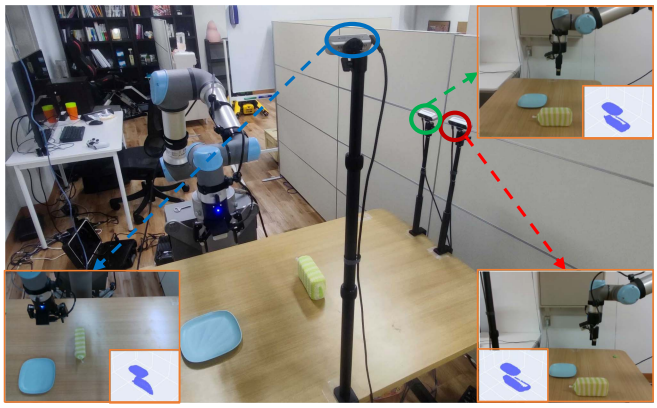

Fig. 8. View generalization experiments. The policy is trained using demonstrations from the camera in the red circle and tested with cameras in the green and blue circles. Despite changes in viewpoint, the affordance point clouds remain similar, enabling the affordance-centric policy to maintain stable performance and reliably execute tasks.

novel or cluttered scenes containing multiple similar objects. As illustrated in Fig. 7, training is performed on a scene with only one striped sponge and one plate present on the desktop. During testing, new task instruction ("pick up the yellow sponge into the pink plate") is provided in a cluttered scene with multiple objects. In this case, VLMs detect several similar objects, and multimodal LLMs successfully select the correct object, leading to successful task execution. In contrast, the scene-level 3D Diffusion Policy shows reasonable tolerance to minor appearance and instance variations but exhibits a significant performance decline in cluttered settings, owing to its limited fine-grained affordance understanding.

3) View Generalization: Fig. 8 shows the view generalization capability of TARAD. The policy is trained using demonstrations collected from the camera in the red circle and tested with cameras in the green and blue circles. Minor changes in camera position result in nearly unchanged performance, whereas significant viewpoint shifts lead to a slight decline in performance. Nonetheless, TARAD remains capable of successfully completing tasks.

This robustness arises from our affordance-centric approach, in which the policy utilizes affordance point clouds as the input, rather than RGB images or full-scene point clouds in 3D Diffusion Policy. Because affordance point clouds focus solely on task-relevant regions, variations in camera viewpoint introduce minimal differences in the input, thereby ensuring consistent and reliable policy execution even under viewpoint shifts.

V. Discussion

We presented TARAD, a novel framework that generates robot manipulation trajectories from natural language instructions and distills precise and generalizable diffusion policies. Our experiments in both simulated and real-world environments demonstrate promising performance; however, several limitations warrant further investigation.

First, it relies on LLMs and VLMs to extract affordance information, which requires reliable object detection. This dependency poses challenges for tasks such as removing objects from closed drawers. Moreover, the current detection capabilities of VLMs are limited, and even when cross-validated with multi-modal LLMs, these constraints can adversely affect system performance.

Second, the actions generated by our method focus exclusively on the end-effector pose, whereas the whole-arm motion is derived via inverse kinematics. Consequently, although the end-effector position can be accurately determined, an inverse kinematics solution can lead to suboptimal or implausible full-arm configurations.

Third, in the data generation stage, our voxel-based affordance representation may be insufficient for tasks that require extremely high precision. For instance, performing fine-grained tasks such as opening a drawer with a small handle in real-world scenarios has proven challenging because of the limitations of the voxel value maps.

Finally, although our automatically generated dataset includes text labels for all sub-tasks, we have not yet evaluated the robot's ability to reuse sub-skills for combinatorial generalization. The integration of foundation models with learning-based policies for robotic manipulation remains in its early stages. Our study represents an initial exploration of this area. We believe that addressing these limitations will be the key to further advancing the state-of-the-art in robot manipulation.

VI. Conclusion

In this work, we introduce TARAD, a framework that integrates LLMs and VLMs to enable zero-shot generation of robotic manipulation trajectories from natural language instructions for diffusion policy training. Our approach highlights commonsense reasoning and the extensive world knowledge embedded in LLMs to generate step-by-step task plans. Subsequently, LLMs and VLMs extract object affordances from RGB-D observations, effectively grounding high-level task instructions in the physical world through contextual understanding and spatial relationships.

By utilizing a heuristic motion planner to generate low-level action plans with affordance as the input, TARAD eliminates the need for predefined motion primitives, thereby facilitating robust and flexible zero-shot trajectory generation.

The integration of affordance information into a 3D diffusion policy, in which affordance serves as the observation space, further enhances the generalization capabilities of the system. Trained on a minimal set of automatically synthesized demonstrations, our imitation learning-based policy not only enables precise manipulation but also demonstrates exceptional adaptability to unseen environments and novel object instances.

Empirical evaluations in simulations and real-world environments have highlighted the effectiveness of combining imitation learning with foundation models. This synergy allows TARAD to distill complex multi-task vision-language policies without relying on expert demonstrations or predefined motion primitives.

REFERENCES

[1] Y. Chen et al., "GravMAD: Grounded spatial value maps guided action diffusion for generalized 3D manipulation," in *Proc. 13th Int. Conf. Learn. Representations*, 2025.

[2] S. Hu, T. Horii, and T. Nagai, "Adaptive and transparent decision-making in autonomous robots through graph-structured world models," *Adv. Robot.*, vol. 38, no. 22, pp. 1579–1599, 2024.

[3] B. Zitkovich et al., "RT-2: Vision-language-action models transfer Web knowledge to robotic control," in *Proc. Conf. Robot Learn.*, PMLR, 2023, pp. 2165–2183.

[4] J. Zhang et al., "SAM-E: Leveraging visual foundation model with sequence imitation for embodied manipulation," in *Proc. 41st Int. Conf. Mach. Learn.*, 2024, pp. 58579–58598.

[5] S. Bahl, R. Mendonca, L. Chen, U. Jain, and D. Pathak, "Affordances from human videos as a versatile representation for robotics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 13778–13790.

[6] C.-C. Hsu, Z. Jiang, and Y. Zhu, "Ditto in the house: Building articulation models of indoor scenes through interactive perception," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 3933–3939.

[7] Y. Tang et al., "UAD: Unsupervised affordance distillation for generalization in robotic manipulation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2025.

[8] J. Achiam et al., "GPT-4 Technical Report," 2023, *arXiv:2303.08774*.

[9] A. Brohan et al., "Do as i can, not as i say: Grounding language in robotic affordances," in *Proc. Conf. Robot Learn.*, PMLR, 2023, pp. 287–318.

[10] K. Lin, C. Agia, T. Migimatsu, M. Pavone, and J. Bohg, "Text2Motion: From natural language instructions to feasible plans," *Auton. Robots*, vol. 47, no. 8, pp. 1345–1365, 2023.

[11] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 9118–9147.

[12] H. Huang, F. Lin, Y. Hu, S. Wang, and Y. Gao, "CoPa: General robotic manipulation through spatial constraints of parts with foundation models," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 9488–9495.

[13] Y. Jin et al., "RobotGPT: Robot manipulation learning from chatGPT," *IEEE Robot. Automat. Lett.*, vol. 9, no. 3, pp. 2543–2550, Mar. 2024.

[14] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "VoxPoser: Composable 3D value maps for robotic manipulation with language models," in *Proc. Conf. Robot Learn.*, PMLR, 2023, pp. 540–562.

[15] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, "RLbench: The robot learning benchmark & learning environment," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 3019–3026, Apr. 2020.

[16] D. Wang, R. Walters, X. Zhu, and R. Platt, "Equivariant q learning in spatial action spaces," in *Proc. Conf. Robot Learn.*, 2022, pp. 1713–1723.

[17] Y. Zhu, A. Joshi, P. Stone, and Y. Zhu, "VIOLA: Imitation learning for vision-based manipulation with object proposal priors," in *Proc. Conf. Robot Learn.*, 2023, pp. 1199–1210.

[18] S. Chen, R. G. Pinel, C. Schmid, and I. Laptev, "PolarNet: 3D point clouds for language-guided robotic manipulation," in *Proc. Conf. Robot Learn.*, PMLR, 2023, pp. 1761–1781.

[19] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in*Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 6840–6851.

[20] H. Ha, P. Florence, and S. Song, "Scaling up and distilling down: Language-guided robot skill acquisition," in *Proc. Conf. Robot Learn.* 2023, pp. 3766–3777.

[21] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3D Diffusion Policy: Generalizable visuomotor policy learning via simple 3D representations," in *Proc. Robot.: Sci. Syst.*, 2024.

[22] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki, "3D diffuser actor: Policy diffusion with 3D scene representations," in *Proc. Conf. Robot Learn.*, PMLR, 2025, pp. 1949–1974.

[23] Y. Ze et al., "GNFactor: Multi-task real robot learning with generalizable neural feature fields," in *Proc. Conf. Robot Learn.* 2023, pp. 284–301.

[24] J. J. Gibson, "The theory of affordances:1979," in *The People, Place, and Space Reader*. Oxfordshire, U.K.: Routledge, 2014, pp. 56–60.

[25] X. Yang, Z. Ji, J. Wu, and Y.-K. Lai, "Recent advances of deep robotic affordance learning: A reinforcement learning perspective," *IEEE Trans. Cogn. Devel. Syst.*, vol. 15, no. 3, pp. 1139–1149, Sep. 2023.

[26] T.-T. Do, A. Nguyen, and I. Reid, "AffordanceNet: An end-to-end deep learning approach for object affordance detection," in *Proc. 2018 IEEE Int. Conf. Robot. Automat.*, 2018, pp. 5882–5889.

[27] D. Turpin, L. Wang, S. Tsogkas, S. Dickinson, and A. Garg, "Gift: Generalizable interaction-aware functional tool affordances without labels," *Robot.: Sci. Syst. XVII*, Robotics: Science and Systems Foundation, 2021.

[28] S. Liu et al., "Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2024, pp. 38–55.

[29] N. Ravi et al., "SAM 2: Segment anything in images and videos," in *Proc. 13th Int. Conf. Learn. Representations*, 2025.

[30] S. Sharan et al., "Plan diffuser: Grounding LLM planners with diffusion models for robotic manipulation," in *Proc. Bridging Gap between Cogn. Sci. Robot Learn. Real World: Progresses New Directions, CoRL 2023 Workshop*, 2024.

[31] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei, "Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation," in *Proc. Conf. Robot Learn.*, PMLR, 2025, pp. 4573–4602.

[32] L. Wang et al., "GenSim: Generating robotic simulation tasks via large language models," in *Proc. 12th Int. Conf. Learn. Representations*, 2024.

[33] P. Hua et al., "GenSim2: Scaling robot data generation with multi-modal and reasoning LLMs," in *Proc. Conf. Robot Learn.*, PMLR, 2025, pp. 5030–5066.

[34] C. Chi et al., "Diffusion policy: Visuomotor policy learning via action diffusion," *Int. J. Robot. Res.*, Oct. 2024, doi: 10.1177/02783649241273668.

[35] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.

[36] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, 2018, pp. 3942–3951.

[37] T. Gervet and Z. Xiao, "Act3D: 3D feature field transformers for multi-task robotic manipulation," in *Proc. Conf. Robot Learn.*, PMLR, 2023, pp. 3949–3965.